

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

L Number	Hits	Search Text	DB	Time stamp
-	871	707/103\$1.ccls.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/12 13: 42
-	4	domain adj value adj set	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/10 15: 30
-	11	(class adj hierarchy) and ((upper lower) adj level adj class)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/09 12: 22
-	1	"5832495".PN.	USPAT; US-PGPUB	2003/07/09 12: 56
-	1	"20030088569"	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/09 15: 31
-	1	hierarchical adj product adj database	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/09 15: 32
-	1	hierarchical adj2 (product adj database)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/09 15: 33
-	609	hierarchical adj database	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/09 15: 35
-	170	(hierarchical adj database) and object and attribute and class	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/09 15: 35
-	1	"6574635"	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/06 15: 12
-	3	6397221.URPN.	USPAT	2004/08/09 09: 30
-	2	("4930071"   "5832495").PN.	USPAT	2004/08/09 09: 31
-	4	hierarch\$ adj product adj catalog	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 09: 35
-	35	hierarch\$ with ( product adj catalog)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 09: 46
-	142	(hierarch\$ and object).ti.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 09: 47
-	92	((hierarch\$ and object).ti.) and (@rlad<=20010509 @ad<=20010509)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 09: 46
-	9	("5361349"   "5465362"   "5581765"   "5588104"   "5603019"   "5649139"   "5794232"   "5802296"   "5991763").PN.	USPAT	2004/08/09 09: 57
-	4	6199059.URPN.	USPAT	2004/08/09 09: 58
-	6	upper adj level adj class	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 10: 36
-	43	lower adj level adj class	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 10: 36
-	34	(lower adj level adj class) and (@rlad<=20010509 @ad<=20010509)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 11: 40
-	1673	(class adj hierarchy) and (@rlad<=20010509 @ad<=20010509)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 11: 40

-	0	((class adj hierarchy) and (@rlad<=20010509 @ad<=20010509)) and (domain adj value adj set)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 11:40
-	153	((class adj hierarchy) and (@rlad<=20010509 @ad<=20010509)) and (domain adj2 set)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 11:49
-	22	((class adj hierarchy) and (@rlad<=20010509 @ad<=20010509)) and superced\$ and inherit\$	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/10 13:55
-	1	hierarchical adj product adj database	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 13:42
-	26	customer adj service adj software	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 13:46
-	38	customer adj service adj computer	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 13:50
-	7	customer adj service adj script	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 14:39
-	0	caseadvisor	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/09 14:39
-	973	707/103R.ccls.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/10 13:55
-	886	707/103R.ccls. and (@rlad<=20010509 @ad<=20010509)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/10 15:27
-	6	domain adj value adj set	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/10 15:28
-	6	domain adj value adj set	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/10 15:29
-	30	hierarchical adj organizational adj structure	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 09:45
-	9	("4965763" "5237502" "5386556" "6061646" "6078917" "6259969" "6314410" "6363301" "6532401" "2002/0156551").PN.	USPAT	2004/08/11 09:32
-	10	("4870579" "4945475" "4996642" "5230072" "5535382" "5583763" "5615341" "5845270" "5940821" "5987454").PN.	USPAT	2004/08/11 09:41
-	1095	705/27.ccls.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 09:45
-	202	705/27.ccls. and hierarch\$	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 09:45
-	161	(705/27.ccls. and hierarch\$) and (@rlad<=20010509 @ad<=20010509)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 11:24
-	26	(overriding adj attribute) and (@rlad<=20010509 @ad<=20010509)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 14:58
-	8	6014637.URPN.	USPAT	2004/08/11 11:46
-	26	("4943932" "5057996" "5101364" "5119475" "5181162" "5195172" "5226161" "5247693" "5249270" "5257384" "5261080" "5274572" "5276775" "5287447" "5293470" "5297283" "5315703" "5367633" "5369766" "5379430" "5388264" "5390325" "5396626" "5398336" "5710887" "5740425").PN.	USPAT	2004/08/11 13:38

-	6	6052670.URPN.	USPAT	2004/08/11 13:38
-	12	5897639.URPN.	USPAT	2004/08/11 13:51
-	31	((overriding near attribute) and (@rlad<=20010509 @ad<=20010509))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 15:07
-	103	((overrid\$ adj attribute) and (@rlad<=20010509 @ad<=20010509))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 16:20
-	67	((overrid\$ adj attribute) and (@rlad<=20010509 @ad<=20010509)) and hierarch\$	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 15:47
-	161	object adj oriented adj data adj model	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 15:48
-	141	((object adj oriented adj data adj model) and (@rlad<=20010509 @ad<=20010509))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 16:03
-	37	((object adj oriented adj data adj model) and (@rlad<=20010509 @ad<=20010509)) and overrid\$	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 16:19
-	62	inheritance with overriding	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 16:20
-	51	((inheritance with overriding) and (@rlad<=20010509 @ad<=20010509))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 16:45
-	3	((overriding adj inheritance) and (@rlad<=20010509 @ad<=20010509))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/11 16:46
-	21	amazon.as.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/12 13:43
-	43	("amazon.com").as.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/08/12 13:43



Terms used **overriding inheritance**

Found 10 of 140,980

Sort results by

relevance



[Save results to a Binder](#)

[Try an Advanced Search](#)

[Try this search in The ACM Guide](#)

Display results

expanded form



[Search Tips](#)

☐ Open results in a new window

Results 1 - 10 of 10

Relevance scale ☐ ☐ ☐ ☐ ☐

## 1 [Inference in DATR](#)

Roger Evans, Gerald Gazdar

April 1989 **Proceedings of the fourth conference on European chapter of the Association for Computational Linguistics**

Full text available: pdf(431.26 KB)



Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)



**DATR** is a declarative language for representing a restricted class of inheritance networks, permitting both multiple and default inheritance. The principal intended area of application is the representation of lexical entries for natural language processing, and we use examples from this domain throughout. In this paper we present the syntax and inference mechanisms for the language. The goal of the **DATR** enterprise is the design of a simple language that (i) has the necessary expressive power ...

## 2 [An object oriented testing and maintenance environment](#)

David C. Kung, Jerry Gao, Pei Hsia

October 1994 **Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(1.37 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)



The object-oriented (OO) paradigm enjoys increasing acceptance in the software industry. Although the OO paradigm has visible benefits in the development cycle, testing and maintenance of OO programs have been considered challenging tasks by the research community. In this paper, we describe an OO testing and maintenance model and present a supporting CASE environment. The model consists of three types of diagrams: the Object Relation Diagram (ORD), Block Branch Diagram (BBD), and Object State Diagram ...

## 3 [The berkeley UNIX consultant project](#)

Robert Wilensky, David N. Chin, Marc Luria, James Martin, James Mayfield, Dekai Wu

December 1988 **Computational Linguistics**, Volume 14 Issue 4

Full text available: pdf(4.41 MB)

[Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)




UC (UNIX Consultant) is an intelligent, natural language interface that allows naive users to learn about the UNIX<sup>2</sup> operating system. UC was undertaken because the task was thought to be both a fertile domain for artificial intelligence (AI) research and a useful application of AI work in planning, reasoning, natural language processing, and knowledge representation. The current implementation of UC comprises the following components: a language analyzer, called ALANA, produces a representation ...

## 4

### [Improving the granularity of access control for Windows 2000](#)

Michael M. Swift, Anne Hopkins, Peter Brundrett, Cliff Van Dyke, Praerit Garg, Shannon Chan,



Full text available:  pdf(447.78 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)


This article presents the mechanisms in Windows 2000 that enable fine-grained and centrally managed access control for both operating system components and applications. These features were added during the transition from Windows NT 4.0 to support the Active Directory, a new feature in Windows 2000, and to protect computers connected to the Internet. While the access control mechanisms in Windows NT are suitable for file systems and applications with simple requirements, they fall short of the ...

**Keywords:** Access control lists, Microsoft Windows 2000, Windows NT, active directory

5 Part II: Articles: Except for exception handling ...

Alexander Romanovsky, Bo Sandén

September 2001 **ACM SIGAda Ada Letters**, Volume XXI Issue 3


Full text available:  pdf(660.46 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Exception handling in Ada has a number of well-known problems. It allows for the propagation of unhandled and anonymous exceptions, it is error-prone and it is inappropriate for some language features such as tasking and tagged types. Ada programs with exceptions can be difficult to understand, develop, modify and analyse, and the exception handling features can be misused in a number of ways. In this paper we introduce the requirements for good exception handling features. We classify the problem ...

6 CommonLoops: merging Lisp and object-oriented programming

Daniel G. Bobrow, Kenneth Kahn, Gregor Kiczales, Larry Masinter, Mark Stefik, Frank Zdybel

June 1986 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications**, Volume 21 Issue 11

Full text available:  pdf(1.06 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

CommonLoops blends object-oriented programming smoothly and tightly with the procedure-oriented design of Lisp. Functions and methods are combined in a more general abstraction. Message passing is invoked via normal Lisp function call. Methods are viewed as partial descriptions of procedures. Lisp data types are integrated with object classes. With these integrations, it is easy to incrementally move a program between the procedure and object-oriented styles.

7 Control principles and role hierarchies

Jonathan D. Moffett


October 1998 **Proceedings of the third ACM workshop on Role-based access control**

Full text available:  pdf(767.83 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

8 Evolution of object behavior using context relations

Linda M. Seiter, Jens Palsberg, Karl J. Lieberherr

October 1996 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 4th ACM SIGSOFT symposium on Foundations of software engineering**, Volume 21 Issue 6

Full text available:  pdf(1.18 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


A collection of design patterns was described by Gamma, Helm, Johnson, and Vlissides in 1994. Recognizing that designs change, each pattern ensures that a certain system aspect can vary over time such as the operations that can be applied to an object or the algorithm of a method. The patterns are described by constructs such as the inheritance and reference relations, attempting to emulate more dynamic relationships. As a result, the design patterns demonstrate how awkward it is to program naturally ...

9 Logical foundations of object-oriented and frame-based languages



Michael Kifer, Georg Lausen, James Wu

July 1995 **Journal of the ACM (JACM)**, Volume 42 Issue 4

Full text available:  pdf(7.52 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

We propose a novel formalism, called Frame Logic (abbr., F-logic), that accounts in a clean and declarative fashion for most of the structural aspects of object-oriented and frame-based languages. These features include object identity, complex objects, inheritance, polymorphic types, query methods, encapsulation, and others. In a sense, F-logic stands in the same relationship to the object-oriented paradigm as classical predicate calculus stands to relational programming. ...

**Keywords:** deductive databases, frame-based languages, logic programming, nonmonotonic inheritance, object-oriented programming, proof theory, semantics, typing

10 Foundations of object-oriented languages



Andrew Black, Jens Palsberg

March 1994 **ACM SIGPLAN Notices**, Volume 29 Issue 3

Full text available:  pdf(617.01 KB)

Additional Information: [full citation](#), [citations](#), [index terms](#)

Results 1 - 10 of 10

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



Welcome  
United States Patent and Trademark Office



» Search

[Help](#) [FAQ](#) [Terms](#) [IEEE Peer Review](#)
**Quick Links****Welcome to IEEE Xplore®**

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

**Tables of Contents**

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

**Search**

- ☐ By Author
- ☐ Basic
- ☐ Advanced

**Member Services**

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

**IEEE Enterprise**

- ☐ Access the IEEE Enterprise File Cabinet

Print Format

Your search matched **8** of **1060766** documents.

A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

**Refine This Search:**

You may refine your search by editing the current search expression or entering new one in the text box.


☐ Check to search within this result set
**Results Key:**

**JNL** = Journal or Magazine   **CNF** = Conference   **STD** = Standard

**1 Calculating attribute values using inheritance structures in fuzzy object oriented data models**

*Pasi, G.; Yager, R.R.;*

Systems, Man and Cybernetics, Part C, IEEE Transactions on , Volume: 29 , Issue 4 , Nov. 1999

Pages:556 - 565

[\[Abstract\]](#)   [\[PDF Full-Text \(132 KB\)\]](#)   **IEEE JNL**

**2 An order-based composition semantics for the inheritance in logic-based object-oriented paradigm**

*Zhi Jin;*

Technology of Object-Oriented Languages, 1997. TOOLS 24. Proceedings , Sept. 1997

Pages:110 - 120

[\[Abstract\]](#)   [\[PDF Full-Text \(492 KB\)\]](#)   **IEEE CNF**

**3 Algebraic semantics of object-oriented data models**

*Zamulin, A.V.;*

Technology of Object-Oriented Languages, 1997. TOOLS 24. Proceedings , Sept. 1997

Pages:43 - 52

[\[Abstract\]](#)   [\[PDF Full-Text \(688 KB\)\]](#)   **IEEE CNF**

**4 A logical foundation for deductive object-oriented databases**

*Liu, M.; Dobbie, G.; Tok Wang Ling;*

Database Systems for Advanced Applications, 2001. Proceedings. Seventh International Conference on , 18-21 April 2001

Pages:116 - 123

[\[Abstract\]](#)   [\[PDF Full-Text \(644 KB\)\]](#)   **IEEE CNF**

**5 Overview of the ROL2 deductive object-oriented database system**



*Mengchi Liu;*

Technology of Object-Oriented Languages and Systems, 1999. TOOLS 30.

Proceedings , 1-5 Aug. 1999

Pages:63 - 72

[\[Abstract\]](#) [\[PDF Full-Text \(192 KB\)\]](#) [IEEE CNF](#)

---

**6 An approach to compute default attribute values in fuzzy object oriented data models**

*Pasi, G.; Yager, R.R.;*

Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on , Volume: 2 , 4-9 May 1998

Pages:1326 - 1331 vol.2

[\[Abstract\]](#) [\[PDF Full-Text \(668 KB\)\]](#) [IEEE CNF](#)

---

**7 COMNET III: object-oriented network performance prediction**

*Jones, J.;*

Simulation Conference Proceedings, 1995. Winter , 3-6 Dec. 1995

Pages:545 - 547

[\[Abstract\]](#) [\[PDF Full-Text \(372 KB\)\]](#) [IEEE CNF](#)

---

**8 A deductive object-oriented language for integrated genome database:**

*Goto, S.; Sakamoto, N.; Takagi, T.;*

System Sciences, 1994. Vol.V: Biotechnology Computing, Proceedings of the Twenty-Seventh Hawaii International Conference on , Volume: 5 , 4-7 Jan. 1994

Pages:108 - 109

[\[Abstract\]](#) [\[PDF Full-Text \(168 KB\)\]](#) [IEEE CNF](#)

---



Terms used **overriding attribute**

Found 7 of 140,980

Sort results by

relevance



[Save results to a Binder](#)

[Try an Advanced Search](#)

[Try this search in The ACM Guide](#)

Display results

expanded form



[Search Tips](#)

☐ Open results in a new window

Results 1 - 7 of 7

Relevance scale ☐ ☐ ☐ ☐ ☐

## 1 [Alias analysis on type inference for class hierarchy in Java](#)

Jongwook Woo, Isabelle Attali, Denis Caromel, Jean-Luc Gaudiot, Andrew L. Wendelborn

January 2001 **Australian Computer Science Communications , Proceedings of the 24th Australasian conference on Computer science**, Volume 23 Issue 1

Full text available: pdf(815.70 KB)

[Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The integration of alias analysis with type information increases the precision of alias detection, especially for inheritance among classes. This paper presents a compile-time flow-sensitive context-insensitive alias analysis algorithm with type information for Java. First, we propose an aliased element representation for an object to compute aliases efficiently. Second, the algorithm computes aliases for shadowed variables by regarding constructors as functions. Third, it performs type inferen ...

## 2 [Object orientation and Fortran 2002: part II](#)

Malcolm Cohen

April 1999 **ACM SIGPLAN Fortran Forum**, Volume 18 Issue 1

Full text available: pdf(380.33 KB) Additional Information: [full citation](#), [citations](#), [index terms](#)

## 3 [An approach to support automatic generation of user interfaces](#)

Prasun Dewan, Marvin Solomon

October 1990 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 12 Issue 4

Full text available: pdf(3.55 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In traditional interactive programming environments, each application individually manages its interaction with the human user. The result is duplication of effort in implementing user interface code and nonuniform—hence confusing—input conventions. This paper presents an approach to support automatic generation of user interfaces in environments based on algebraic languages. The approach supports the editing model of interaction, which allows a user to view all appli ...

## 4 [Transformations on a dialog tree: rule-based mapping of content to style](#)

W. E. Bennett, S. J. Boies, J. D. Gould, S. L. Greene, C. F. Wiecha

November 1989 **Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology**

Full text available: pdf(1.00 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

## 5 [XML transactions: An object-oriented extension of XML for autonomous web](#)

applications

Hasan M. Jamil, Giovanni A. Modica

November 2002 **Proceedings of the eleventh international conference on Information and knowledge management**

Full text available:  pdf(277.52 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

While the idea of extending XML to include object-oriented features has been gaining popularity in general, the potential of inheritance in document design has not been well recognized in contemporary research. In this paper we demonstrate that XML with dynamic inheritance aids better document designs and decreased management overheads and support increased autonomy. As an extended application, we point out that dynamic inheritance also helps effective automated web portal and ontology designs.W ...

**Keywords:** XML, autonomous objects, document structuring, dynamic object hierarchy, inheritance, object-orientation, web

6 [Incorporating visual design into information systems courses: a practical primer reshaping the human, computer interface design curriculum](#)

John Campbell Finnegan, Jeffrey A. Griffin

October 2000 **Journal of Computing Sciences in Colleges**, Volume 16 Issue 1

Full text available:  pdf(383.09 KB) Additional Information: [full citation](#), [references](#), [index terms](#)

7 [Generating hypermedia from specifications by sketching multimedia templates](#)

S. Fraïssé, J. Nanard, M. Nanard

February 1997 **Proceedings of the fourth ACM international conference on Multimedia**

Full text available:  pdf(1.42 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

**Keywords:** executable specification, hypermedia design, hypermedia generation, multimedia template, object-oriented

Results 1 - 7 of 7

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



Terms used **class hierarchy** and **domain value**

Found 7 of 140,980

Sort results  
by

Display  
results

[Save results to a Binder](#)

[Search Tips](#)

☐ [Open results in a new window](#)

[Try an Advanced Search](#)

[Try this search in The ACM Guide](#)

Results 1 - 7 of 7

Relevance scale ☐ ☐ ☐ ☐ ☐

## 1 [H-trees: a dynamic associative search index for OODB](#)

Chee Chin Low, Beng Chin Ooi, Hongjun Lu

June 1992 **ACM SIGMOD Record , Proceedings of the 1992 ACM SIGMOD international conference on Management of data**, Volume 21 Issue 2

Full text available: [pdf\(1.06 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The support of the superclass-subclass concept in object-oriented databases (OODB) makes an instance of a subclass also an instance of its superclass. As a result, the access scope of a query against a class in general includes the access scope of all its subclasses, unless specified otherwise. To support the superclass-subclass relationship efficiently, the index must achieve two objectives. First, the index must support efficient retrieval of instances from a single class. Second, it must ...

## 2 [Logical foundations of object-oriented and frame-based languages](#)

Michael Kifer, Georg Lausen, James Wu

July 1995 **Journal of the ACM (JACM)**, Volume 42 Issue 4

Full text available: [pdf\(7.52 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

We propose a novel formalism, called Frame Logic (abbr., F-logic), that accounts in a clean and declarative fashion for most of the structural aspects of object-oriented and frame-based languages. These features include object identity, complex objects, inheritance, polymorphic types, query methods, encapsulation, and others. In a sense, F-logic stands in the same relationship to the object-oriented paradigm as classical predicate calculus stands to relational programming. ...

**Keywords:** deductive databases, frame-based languages, logic programming, nonmonotonic inheritance, object-oriented programming, proof theory, semantics, typing

## 3 [A framework for the management of past experiences with time-extended situations](#)

Michel Jaczynski

January 1997 **Proceedings of the sixth international conference on Information and knowledge management**

Full text available: [pdf\(1.20 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

## 4 [A configurable type hierarchy index for OODB](#)

Thomas A. Mueck, Martin L. Polaschek

November 1997 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 6 Issue 4

Full text available: [pdf\(411.47 KB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)


With respect to the specific requirements of advanced OODB applications, index data structures for type hierarchies in OODBMS have to provide efficient support for multiattribute queries and have to allow index optimization for a particular query profile. We describe the *multikey type index* and an efficient implementation of this indexing scheme. It meets both requirements: in addition to its multiattribute query capabilities it is designed as a mediator between two standard design altern ...

**Keywords:** Access methods, Indexing, Multiple inheritance, OODB, Type hierarchies

##### 5 Query processing for knowledge bases using join indices

Adel Shrufi, Thodoros Topaloglou

December 1995 **Proceedings of the fourth international conference on Information and knowledge management**

Full text available:  [pdf\(1.03 MB\)](#)


Additional Information: [full citation](#), [references](#), [index terms](#)



##### 6 Data model for extensible support of explicit relationships in design databases

Joan Peckham, Bonnie MacKellar, Michael Doherty

April 1995 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 4 Issue 2

Full text available:  [pdf\(2.01 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)



We describe the conceptual model of SORAC, a data modeling system developed at the University of Rhode Island. SORAC supports both semantic objects and relationships, and provides a tool for modeling databases needed for complex design domains. SORAC's set of built-in semantic relationships permits the schema designer to specify enforcement rules that maintain constraints on the object and relationship types. SORAC then automatically generates C++ code to maintain the specified enforcement rules ...

**Keywords:** computer-aided architectural design, database constraints, relationship semantics, semantic and object-oriented data modeling

##### 7 C++ classes for linking optimization with complex simulations

Mark S. Gockenbach, Matthew J. Petro, William W. Symes

June 1999 **ACM Transactions on Mathematical Software (TOMS)**, Volume 25 Issue 2

Full text available:  [pdf\(131.14 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)



The object-oriented programming paradigm can be used to overcome the incompatibilities between off-the-shelf optimization software and application software. The Hilbert Class Library (HCL) defines the fundamental mathematical objects arising in optimization problems, such as vectors, linear operators, and so forth, as C++ classes, making it possible to write optimization code in a natural fashion, while allowing application software such as simulators to use the most convenient data structures a ...

**Keywords:** object-oriented design, optimization, simulation

Results 1 - 7 of 7

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)